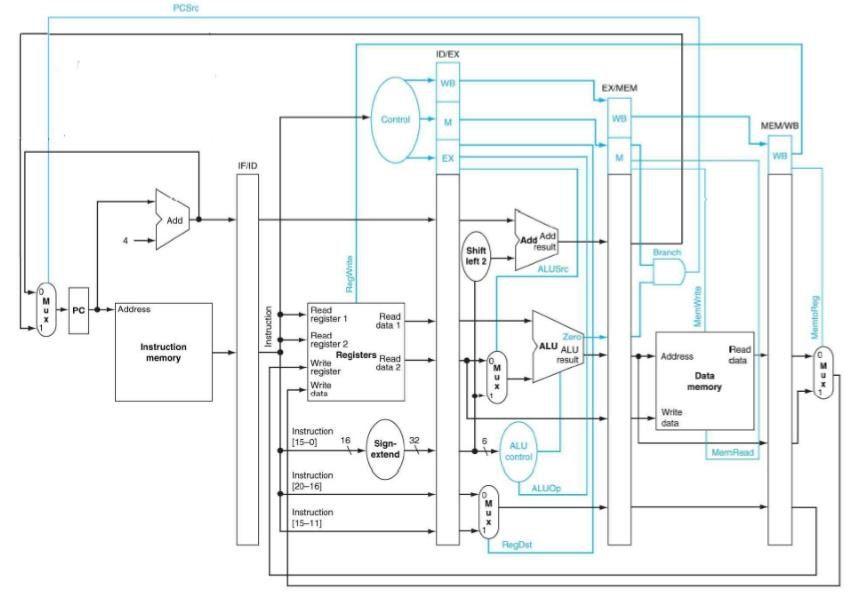
**Computer Organization Lab4**

**0516022 吳禹璇**

Source code and the note:

我沒有覺得什麼地方是比較重要的，只是我寫完之後有聽到有些人在接pipe\_cpu\_1的時候，是把全部signal先合併成一個wire再傳進pipe reg，在使用的時候就要很清楚每一個signal是在第幾個bit到第幾個bit的位置才能呼叫。但是我用的方法是直接在傳入pipe reg的時候把訊號串在一起，傳出的時候也是直接使用新的訊號串在一起，使用上我是覺得比較方便，但是也不知道有沒有比較厲害就是了。

My architecture:



Hardware module analysis:

這次基本都是用之前寫過的，不過因為沒有用到jump那些東西，所以把 jump的control unit刪掉了（branchtype也是，因為用不到），並把一些為了區別jump的control unit從2 bit改回1 bit。

在alu和alu\_ctrl的地方也有做更動，因為多了mult的指令，mult也是r-type，所以不需要再更改decoder，我在alu\_ctrl的地方做mult的判斷，並在alu裡面加了一個乘法的運算。

co\_p4\_test\_2.txt的部分，因為有hazard，我使用的解決方法是reorder instructions，以下是我的更動順序之後的instruction。

00100000000000010000000000010000

00100000000000110000000000001000

00100000000010010000000001100100

00100000001000100000000000000100

10101100000000010000000000000100

10001100000001000000000000000100

00100000001001110000000000001010

00000000011000010011000000100000

00000000100000110010100000100010

00000000111000110100000000100100

附上更改以後的，這邊會比較容易看出我做的更動是哪邊。

I1: addi $1, $0, 16

I3: addi $3, $0, 8

I10: addi $9, $0, 100

I2: addi $2, $1, 4

I4: sw $1, 4($0)

I5: lw $4, 4($0)

I8: addi $7, $1, 10

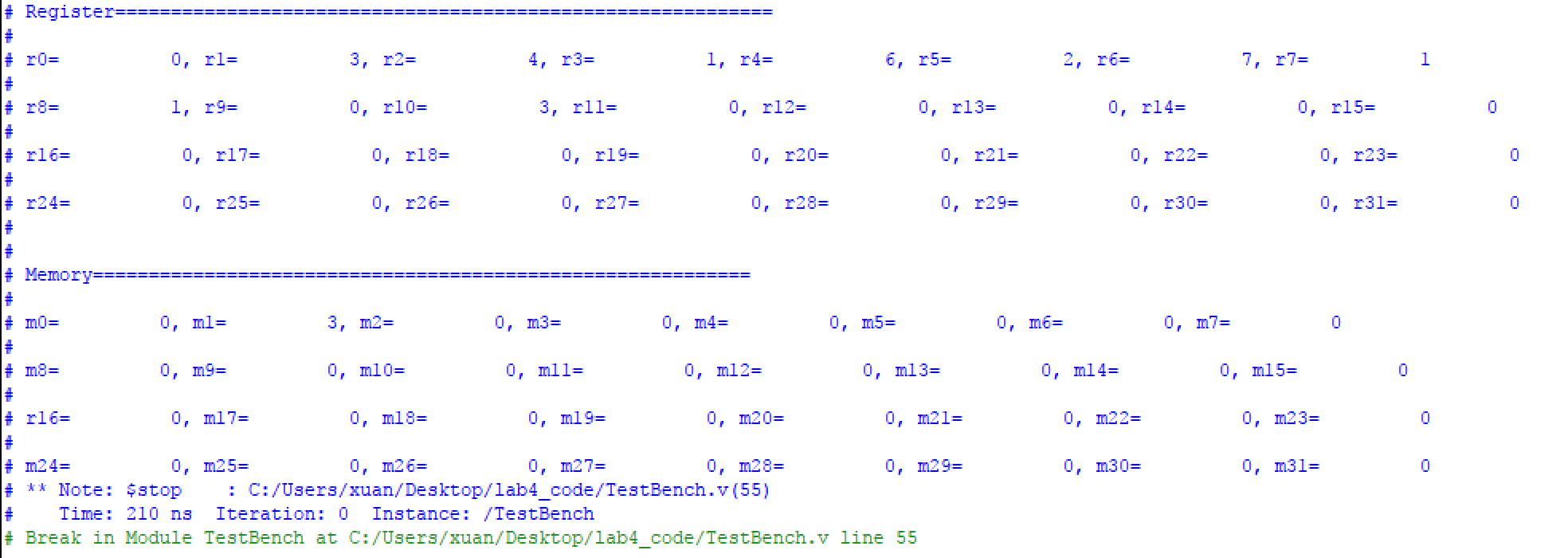
I7: add $6, $3, $1

I6: sub $5, $4, $3

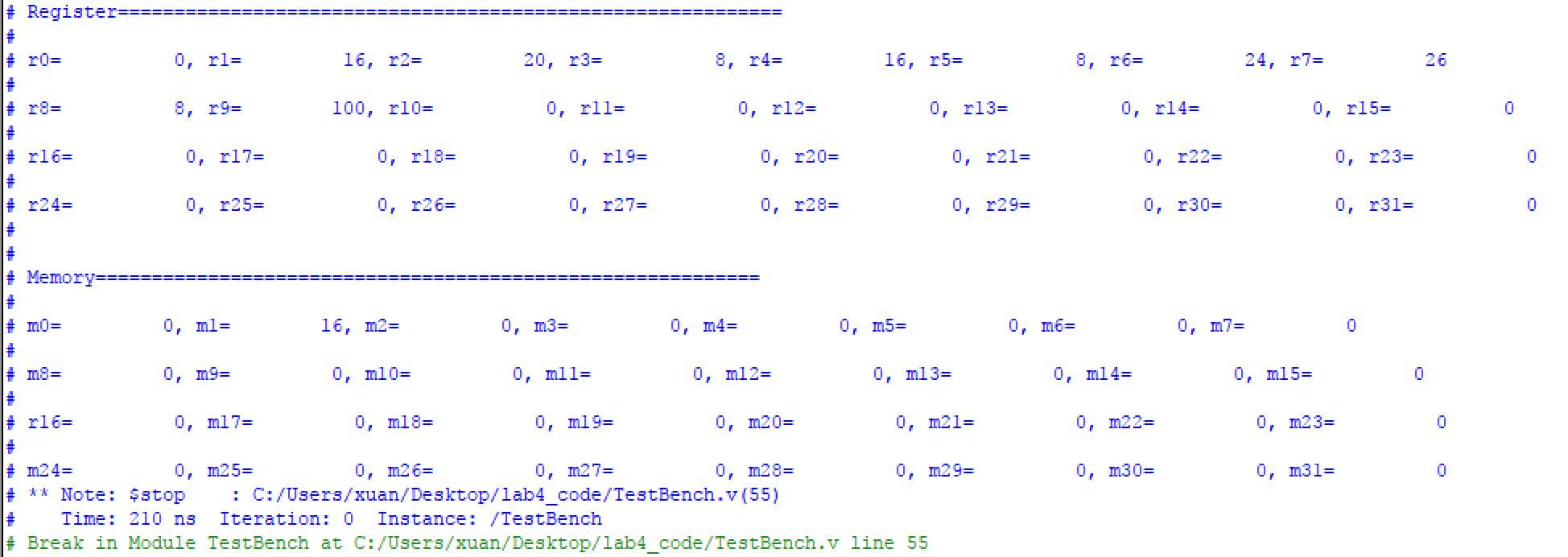
I9: and $8, $7, $3

Finished part:

co\_p4\_test\_1 :



co\_p4\_test\_2 (after modified) :



Problems you met and solutions:

這次比較沒有遇到什麼問題，唯一比較麻煩的是在接pipe register 的時候，因為從前一個stage要進到下一個stage的時候名稱不一樣，要很細心的改到每一個地方，才不會線沒接好跑出錯的答案。不過因為我一開始就有預料到很容易搞混，所以做的時候有一個一個stage做更動，就沒有遇到接錯的問題。

Summary:  
 寫完這次lab對pipeline了解的更透徹，不過因為還沒有做forwarding的部分，所以目前沒有太複雜，但是我猜下一次應該就會要寫那部分了，希望是最後一次作業了QQ。